

A CURVILINEAR SEARCH METHOD FOR P -HARMONIC FLOWS ON SPHERES

DONALD GOLDFARB [†], ZAIWEN WEN [†], AND WOTAO YIN [‡]

December 5, 2007

Abstract. The problem of finding p -harmonic flows arises in a wide range of applications including micromagnetics, liquid crystal theory, directional diffusion, and chromaticity denoising. In this paper, we propose an innovative curvilinear search method for minimizing p -harmonic energies over spheres. Starting from a flow (map) on the unit sphere, our method searches along a curve that lies on the sphere in a manner similar to a standard inexact line search descent method. We show that our method is globally convergent if the step length satisfies the Armijo-Wolfe conditions. Computational tests are presented to demonstrate the efficiency of the proposed method and a variant of it that uses Barzilai-Borwein steps.

Key words. energy minimization, p -harmonic maps, p -harmonic flows, finite difference, curvilinear search, global convergence, chromaticity denoising

AMS subject classifications. 58E20, 65N06, 65N12, 65M06, 68U10, 90C26, 90C30, 94A08

1. Introduction. Let $|\cdot|$ denote the Euclidean norm, S^{N-1} denote the $(N-1)$ -sphere in \mathbb{R}^N , i.e., $S^{N-1} := \{\mathbf{U} \in \mathbb{R}^N : |\mathbf{U}| = 1\}$, and Ω denote an open subset of \mathbb{R}^M , where $M \geq 1$ and $N \geq 2$. In this paper we propose, for the case $N = 3$ and $M = 2$, simple and very efficient methods for solving the minimization problem

$$(1.1) \quad \min_{\mathbf{U} \in H_{\mathbf{n}_0}^1(\Omega, S^{N-1})} \widehat{E}_p(\mathbf{U}) = \int_{\Omega} |\mathcal{D}\mathbf{U}(\mathbf{x})|_F^p \, d\mathbf{x},$$

where $1 \leq p < \infty$,

$$(1.2) \quad H_{\mathbf{n}_0}^1(\Omega, S^{N-1}) := \{\mathbf{U} \in H^1(\Omega, \mathbb{R}^N) \mid \mathbf{U}(\mathbf{x}) \in S^{N-1} \text{ a.e.}; \mathbf{U}|_{\partial\Omega} = \mathbf{n}_0\},$$

$|\cdot|_F$ is the Frobenius norm, i.e., $|B|_F = \sqrt{\sum_{i,j} B_{i,j}^2}$, and the operator \mathcal{D} denotes differentiation, i.e.,

$$\mathcal{D}\mathbf{U}(\mathbf{x}) = \begin{pmatrix} \frac{\partial \mathbf{U}_1(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial \mathbf{U}_1(\mathbf{x})}{\partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial \mathbf{U}_N(\mathbf{x})}{\partial x_1} & \cdots & \frac{\partial \mathbf{U}_N(\mathbf{x})}{\partial x_M} \end{pmatrix}$$

is the Jacobian matrix of \mathbf{U} at \mathbf{x} . In what follows, we use the simpler notation $|\mathbf{U}| = 1$ to represent the constraint $\mathbf{U} \in H_{\mathbf{n}_0}^1(\Omega, S^{N-1})$. The mappings \mathbf{U} that are stationary points of Problem (1.1) are called p -harmonic maps, and in particular, harmonic maps for $p = 2$.

The analytical properties of (1.1), including the existence, non-uniqueness, regularity and singularities of minimizing harmonic maps, have been intensively studied [Chen, 1989; Chen and Lin, 1995; Chen and Struwe, 1989; Coron and Gulliver, 1989;

[†]Department of Industrial Engineering and Operations Research, Columbia University, New York, 10027, U.S.A. (goldfarb@columbia.edu, zw2109@columbia.edu). Research supported in part by NSF Grant DMS 06-06712, ONR Grant N000140310514 and DOE Grant GE-FG01-92ER-25126.

[‡]Department of Computational and Applied Mathematics, Rice University, Texas, 77005, U.S.A. (wotao.yin@rice.edu).

Hungerbühler, 1996, 1997, 2004; Lin, 1987]. The applications of p -harmonic maps include micromagnetics [Kružík and Prohl, 2006], liquid crystal theory [Alouges, 1997; Cohen et al., 1987, 1989; Lin and Luskin, 1989], directional diffusion and chromaticity (color image) denoising [Chan and Shen, 2000; Lysaker et al., 2004; Perona, 1998; Tang et al., 2000, 2001].

Several types of numerical approaches have been proposed to solve Problem (1.1). The approach in [Cohen et al., 1987, 1989] solves the Euler Lagrange equations for (1.1) iteratively, where at each step the spherical constraints are ignored at first and then the solution \mathbf{V} is renormalized by setting $\mathbf{U} = \frac{\mathbf{V}}{|\mathbf{V}|}$. This renormalization approach is analyzed in [Alouges, 1997], where it is shown that the energy is decreased after each renormalized step and a convergent algorithm is proposed. Related finite element methods are studied in [Barrett et al., 2007; Bartels, 2005]. By modifying a discretization scheme for the heat flow equations corresponding to (1.1), constraint preserving finite element methods are developed in [Bartels and Prohl, 2006,c]. A second approach [Bethuel et al., 1992, 1993; Misawa, 2001] adds a penalty term to the objective function in (1.1) to penalize violation of the spherical constraint, i.e., it forms

$$\mathcal{P}_\epsilon(\mathbf{U}) = \int_{\Omega} |\mathcal{D}\mathbf{U}(\mathbf{x})|_F^p \, d\mathbf{x} + \frac{1}{\epsilon} \int_{\Omega} (|\mathbf{U}|^2 - 1)^2 \, d\mathbf{x},$$

and then solves a sequence of unconstrained minimization problems $\min \mathcal{P}_\epsilon(\mathbf{U})$ by letting $\epsilon \rightarrow 0$. This approach is also used to solve the minimization of the Ginzburg-Landau functional. A third approach [Cecil et al., 2004; Vese and Osher, 2002] is based on solving the unconstrained problem

$$(1.3) \quad \min_{\mathbf{U}} E_p(\mathbf{U}) = \int_{\Omega} \left| \mathcal{D} \left(\frac{\mathbf{U}}{|\mathbf{U}|} \right) \right|_F^p \, d\mathbf{x}, \quad \text{s.t. } \mathbf{U} \in H_{\mathbf{n}_0}^1(\Omega, \mathbb{R}^N).$$

A parameterization of the variable \mathbf{U} is employed to derive a constraint preserving gradient descent method.

Our method directly solves (1.1) by generating a sequence $\{\mathbf{U}^n\}$ based upon an updating formula that *preserves* $|\mathbf{U}^n| = 1$ *and does not involve renormalization*. Every update from \mathbf{U}^n to \mathbf{U}^{n+1} is determined by a descent direction on the manifold $H_{\mathbf{n}_0}^1(\Omega, S^{N-1})$ and a step size; however, the formula preserves $|\mathbf{U}^{n+1}| = 1$ for any direction and step size. This important property allows us to directly apply classical optimization techniques developed for use in Euclidean spaces such as line search methods and Barzilai-Borwein step sizes to significantly accelerate convergence, resulting in a framework that we refer to as a *curvilinear method*. In addition, at least for $N \leq 3$, we show that the updating formula is simple and easy to compute. We carefully discretized various quantities, especially the objective functional and updating formula, so that the magnitude preserving property holds after discretization and, therefore, benefits numerical computation.

This paper is organized as follows. In Section 2.1, we derive a descent method that preserves $|\mathbf{U}| = 1$ from the Euler-Lagrange equations for (1.1). In Section 2.2, we present an equivalent representation of the unconstrained objective functional in (1.3) for the case $N = 3$. We then introduce in Section 2.3 a discrete counterpart of the descent method described in Section 2.1. In Section 2.4, we present a curvilinear search method applied to the discrete formulation and prove global convergence to a stationary point. In Section 3 we describe how to incorporate Barzilai-Borwein steps into our curvilinear search framework. Finally a set of numerical results on both

synthetic and real problems are presented in Section 4 to demonstrate the efficiency of our algorithms.

2. Continuous and Discrete Descent Methods.

2.1. A Descent Method that Preserves $|\mathbf{U}^n| = 1$. It is well known that the Euler-Lagrange equations (i.e., the first-order optimality conditions) for Problems (1.1) and (1.3) are the set of coupled PDEs (assuming they are well-defined)

$$(2.1) \quad \begin{cases} \nabla \widehat{E}_p(\mathbf{U}) := -\Delta_p \mathbf{U} - |\mathcal{D}\mathbf{U}|^p \mathbf{U} = 0, \\ |\mathbf{U}| = 1 \text{ a.e. on } \Omega_T, \\ \mathbf{U}|_{\partial\Omega} = \mathbf{n}_0, \end{cases}$$

where $\nabla \widehat{E}_p(\mathbf{U})$ denotes the Fréchet derivative of $\widehat{E}_p(\mathbf{U})$ with respect to \mathbf{U} and $\Delta_p \mathbf{U}^n := \mathcal{D} \cdot (|\mathcal{D}\mathbf{U}|^{p-2} \mathcal{D}\mathbf{U})$. From $|\mathbf{U}| = 1$ it follows that $(\mathcal{D}\mathbf{U})^\top \mathbf{U} = \mathbf{0}$; hence from this and the product rule for differentiation we obtain

$$0 = \mathcal{D} \cdot (|\mathcal{D}\mathbf{U}|^{p-2} (\mathcal{D}\mathbf{U}^\top \mathbf{U})) = \mathcal{D} \cdot ((|\mathcal{D}\mathbf{U}|^{p-2} \mathcal{D}\mathbf{U})^\top \mathbf{U}) = |\mathcal{D}\mathbf{U}|^p + \langle \mathbf{U}, \Delta_p \mathbf{U} \rangle,$$

which implies that $-|\mathcal{D}\mathbf{U}|^p = \langle \mathbf{U}, \Delta_p \mathbf{U} \rangle$ and

$$\nabla \widehat{E}_p(\mathbf{U}) = -|\mathcal{D}\mathbf{U}|^p \mathbf{U} - \Delta_p \mathbf{U} = \langle \mathbf{U}, \Delta_p \mathbf{U} \rangle \mathbf{U} - \langle \mathbf{U}, \mathbf{U} \rangle \Delta_p \mathbf{U} = \mathbf{U} \times (\mathbf{U} \times \Delta_p \mathbf{U}),$$

where the second equality is from Lagrange's formula

$$(2.2) \quad \mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \langle \mathbf{a}, \mathbf{c} \rangle \mathbf{b} - \langle \mathbf{a}, \mathbf{b} \rangle \mathbf{c}.$$

Given a current point \mathbf{U}^n with $|\mathbf{U}^n| = 1$, the classical steepest descent method computes a new trial point $\mathbf{U}_{\text{SD}}^{n+1}$ as

$$(2.3) \quad \mathbf{U}_{\text{SD}}^{n+1} = \mathbf{U}^n - \tau \nabla \widehat{E}_p(\mathbf{U}^n) = \mathbf{U}^n - \tau \mathbf{U}^n \times (\mathbf{U}^n \times \Delta_p \mathbf{U}^n),$$

where τ is a step size. In general, $\mathbf{U}_{\text{SD}}^{n+1}$ does not satisfy $|\mathbf{U}_{\text{SD}}^{n+1}| = 1$. Inspired by [Vese and Osher, 2002], we propose replacing the step direction

$$\mathbf{U}^n \times (\mathbf{U}^n \times \Delta_p \mathbf{U}^n)$$

by

$$\frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \times (\mathbf{U}^n \times \Delta_p \mathbf{U}^n),$$

which yields the following method for computing a new trial point \mathbf{U}^{n+1}

$$(2.4) \quad \mathbf{U}^{n+1} = \mathbf{U}^n - \tau \frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \times (\mathbf{U}^n \times \Delta_p \mathbf{U}^n).$$

We note that both (2.3) and (2.4) can be viewed as a time discretization of p -harmonic heat flow $\mathbf{U}_\tau = -\nabla \widehat{E}_p(\mathbf{U})$ by letting τ go to 0.

REMARK 2.1. *We can represent the cross product in \mathbb{R}^3 in the form of matrix-vector product. Let $\mathbf{a} = (a_1, a_2, a_3)^\top \in \mathbb{R}^3$ and $\mathbf{b} \in \mathbb{R}^3$. Then the cross product $\mathbf{a} \times \mathbf{b} = \mathbf{a}^{(\times)} \mathbf{b}$, where the matrix $\mathbf{a}^{(\times)}$ is defined by*

$$(2.5) \quad \mathbf{a}^{(\times)} = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix}.$$

Using this result, we can obtain \mathbf{U}^{n+1} explicitly.

THEOREM 2.2. *For any \mathbf{H}^n , the solution of the update formula*

$$(2.6) \quad \mathbf{U}^{n+1} = \mathbf{U}^n - \frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \times \mathbf{H}^n,$$

with respect to \mathbf{U}^{n+1} is

$$(2.7) \quad \mathbf{U}^{n+1} = \left(I - \frac{1}{2}(\mathbf{H}^n)^{(\times)} \right)^{-1} \left(I + \frac{1}{2}(\mathbf{H}^n)^{(\times)} \right) \mathbf{U}^n$$

where I is the identity matrix in $\mathbb{R}^{3 \times 3}$ and $(\mathbf{H}^n)^{(\times)}$ is the matrix of the form (2.5) corresponding to \mathbf{H}^n . In addition, (2.7) satisfies

$$(2.8) \quad |\mathbf{U}^{n+1}| = |\mathbf{U}^n|.$$

Consequently, if $|\mathbf{U}^0| = 1$ it follows that $\{\mathbf{U}^n\}$ satisfies $|\mathbf{U}^n| = 1$ for all $n > 0$.

Proof. 1. Rearranging (2.6), we get

$$\mathbf{U}^{n+1} - \mathbf{U}^n + \frac{1}{2}\mathbf{U}^{n+1} \times \mathbf{H}^n + \frac{1}{2}\mathbf{U}^n \times \mathbf{H}^n = 0.$$

Using the matrix-vector form of the cross product (2.5), the fact that $\mathbf{a} \times \mathbf{b} = -\mathbf{b} \times \mathbf{a}$ and collecting common terms, we obtain

$$(2.9) \quad \left(I - \frac{1}{2}(\mathbf{H}^n)^{(\times)} \right) \mathbf{U}^{n+1} = \left(I + \frac{1}{2}(\mathbf{H}^n)^{(\times)} \right) \mathbf{U}^n.$$

Since $(\mathbf{H}^n)^{(\times)}$ is skew symmetric, the matrix $\left(I - \frac{1}{2}(\mathbf{H}^n)^{(\times)} \right)$ is nonsingular for any \mathbf{H}^n ; in fact, $\det \left(I - \frac{1}{2}(\mathbf{H}^n)^{(\times)} \right) = 1 + \frac{1}{4}\|\mathbf{H}^n\|^2$. Therefore, the system of equations (2.9) is solvable and we obtain (2.7).

2. Taking the inner product on both sides of (2.6) with $\frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2}$ and rearranging terms gives

$$\begin{aligned} 0 &= \left\langle \mathbf{U}^{n+1} - \mathbf{U}^n, \frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \right\rangle + \left\langle \frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \times \mathbf{H}^n, \frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \right\rangle \\ &= \frac{1}{2} (|\mathbf{U}^{n+1}|^2 - |\mathbf{U}^n|^2), \end{aligned}$$

where we have used the fact that the second term in the first equation is equal to zero because of the following properties of the cross product: $\langle \mathbf{a} \times \mathbf{b}, \mathbf{c} \rangle = -\langle \mathbf{b}, \mathbf{a} \times \mathbf{c} \rangle$ and $\mathbf{a} \times \mathbf{a} = \mathbf{0}$ for any $\mathbf{a}, \mathbf{b}, \mathbf{c} \in \mathbb{R}^3$. This result also follows from (2.7) and the fact that $B^\top B = I$ if $B = (I - K)^{-1}(I + K)$ and K is skew symmetric. \square

THEOREM 2.3. *The step $\mathbf{U}^{n+1} - \mathbf{U}^n$ in (2.4) is a descent step if the steepest descent step $\mathbf{U}_{\text{SD}}^{n+1} - \mathbf{U}^n$ with the same step size τ is a descent step.*

Proof. It is easy to verify that

$$\mathbf{U}^{n+1} - \mathbf{U}_{\text{SD}}^{n+1} = \frac{\tau}{2}(\mathbf{H}^n)^{(\times)} (\mathbf{U}^{n+1} - \mathbf{U}^n),$$

where $\mathbf{H}^n = \mathbf{U}^n \times \Delta_p \mathbf{U}^n$. Since $(\mathbf{H}^n)^{(\times)}$ is skew symmetric,

$$(\mathbf{U}^{n+1} - \mathbf{U}^n)^\top (\mathbf{U}^{n+1} - \mathbf{U}_{\text{SD}}^{n+1}) = 0.$$

Hence

$$\begin{aligned} (\mathbf{U}^{n+1} - \mathbf{U}^n)^\top (\mathbf{U}_{\text{SD}}^{n+1} - \mathbf{U}^n) &= (\mathbf{U}^{n+1} - \mathbf{U}^n)^\top (\mathbf{U}_{\text{SD}}^{n+1} - \mathbf{U}^{n+1} + \mathbf{U}^{n+1} - \mathbf{U}^n) \\ &= |\mathbf{U}^{n+1} - \mathbf{U}^n|^2 > 0, \end{aligned}$$

since $\mathbf{U}^{n+1} \neq \mathbf{U}^n$ if $\mathbf{U}_{\text{SD}}^{n+1} \neq \mathbf{U}^n$. \square

In [Bartels and Prohl, 2006c] the authors take \mathbf{H}^n as $\frac{\mathbf{U}^{n+1} + \mathbf{U}^n}{2} \times \Delta_p \mathbf{U}^{n+1}$ in a finite element discretization for the gradient flow equation and use a fixed point method to compute \mathbf{U}^{n+1} . Similar results are also presented in [Bartels and Prohl, 2006b,a]

REMARK 2.4. *The updating formula (2.4) can be applied to solve the general minimization problem*

$$(2.10) \quad \min \mathcal{F}(\mathbf{U}), \quad \text{subject to } \mathbf{U} \in S^2.$$

From the theory of Riemannian manifolds, the Euler-Lagrangian equations of Problem (2.10), given by the covariant derivative, is

$$(2.11) \quad \Pi_{\mathbf{U}}(\nabla \mathcal{F}) = \nabla \mathcal{F} - \langle \mathbf{U}, \nabla \mathcal{F} \rangle \mathbf{U} = 0, \quad \mathbf{U} \in S^2,$$

where $\Pi_{\mathbf{U}}$ is the orthogonal projection from $T_{\mathbf{U}}\mathbb{R}^3$ onto the tangent space $T_{\mathbf{U}}S^2$ [Chan and Shen, 2000; Mémoli et al., 2004]. From the Lagrange formula (2.2), we obtain

$$\Pi_{\mathbf{U}}(\nabla \mathcal{F}) = -\mathbf{U} \times (\mathbf{U} \times \nabla \mathcal{F}), \quad \mathbf{U} \in S^2,$$

which implies that an updating formula similar to (2.4) can be used to solve (2.10).

Next, we develop a discrete numerical scheme that preserves the results of Theorems 2.2 and 2.3 after first presenting an objective functional in next section that is equivalent to the one in (1.3).

2.2. An Equivalent Objective Functional. By expanding the integral $\left| \mathcal{D} \left(\frac{\mathbf{U}}{|\mathbf{U}|} \right) \right|_F^p$ in (1.3) for $\mathbf{U} = (u, v, w)$, we obtain

$$(2.12) \quad \min_{\mathbf{U} \in H_{n_0}^1(\Omega, \mathbb{R}^N)} E_p(\mathbf{U}) = \int_{\Omega} \left(\left| \frac{u\mathcal{D}v - v\mathcal{D}u}{u^2 + v^2 + w^2} \right|^2 + \left| \frac{u\mathcal{D}w - w\mathcal{D}u}{u^2 + v^2 + w^2} \right|^2 + \left| \frac{v\mathcal{D}w - w\mathcal{D}v}{u^2 + v^2 + w^2} \right|^2 \right)^{p/2} dx,$$

which is equivalent to Problems (1.1) and (1.3). Our task is to discretize (2.12) and express the updating formula (2.4) in terms of (u^k, v^k, w^k) while preserving the results in Theorems 2.2 and 2.3.

Although the minimization problems (1.1) and (1.3) (hence, (2.12) above as a special case for $N = 3$) have the same infimum, there are advantages in solving (1.3) (hence, (2.12)) instead of the equivalent problem (1.1). In (1.1) the explicit constraints $|\mathbf{U}| = 1$ are all nonconvex and thus difficult to handle numerically; in contrast problem (1.3) or (2.12) is unconstrained and one can directly apply nonlinear optimization methods to it. For (2.12), we give below a finite difference discretization for which we derive a globally convergent and efficient iterative algorithm. Furthermore, we do not need to renormalize \mathbf{U} , either at every iteration or when our algorithm terminates (at a steady state).

Formula (2.12) can be extended to higher dimensional spaces, i.e., $N \geq 3$, by expanding the integral $\left| \mathcal{D} \left(\frac{\mathbf{U}}{|\mathbf{U}|} \right) \right|_F^p$ explicitly. Also let us remark that (2.12) for $N = 2$,

i.e., when $\mathbf{U} = (u, v)$ maps to S^1 , can be simplified. Needless to say, all formulas and results developed below for $N = 3$ can also be simplified and applied to $N = 2$.

REMARK 2.5. For p -harmonic mappings into S^1 , we simply take $w = 0$; hence Problem (1.3) becomes

$$\min E_p(\mathbf{U}) = \int_{\Omega} \left| \frac{u\mathcal{D}v - v\mathcal{D}u}{u^2 + v^2} \right|^p d\mathbf{x},$$

which has a gradient flow governed by

$$(2.13) \quad u_t + v\mathbf{H} = 0, \quad v_t - u\mathbf{H} = 0,$$

where $\mathbf{H} = u\mathcal{D} \cdot (|\mathcal{D}\mathbf{U}|^{p-2}\mathcal{D}v) - v\mathcal{D} \cdot (|\mathcal{D}\mathbf{U}|^{p-2}\mathcal{D}u)$.

REMARK 2.6. For $N = 2$, i.e., mappings $\mathbf{U} = (u, v)$ into S^1 , [Vese and Osher, 2002] parametrizes u and v by (r, θ) using $u = r \cos(\theta)$ and $v = r \sin(\theta)$. It can be shown that $\left| \mathcal{D} \left(\frac{\mathbf{U}}{|\mathbf{U}|} \right) \right|_F^p \equiv |\mathcal{D}\theta|^p$. The authors calculate the gradient flow of the functional $\int_{\Omega} |\mathcal{D}\theta|^p d\mathbf{x}$ with respect to θ . Using the transformation $\theta = \tan^{-1} \left(\frac{v}{u} \right)$ and the two facts that $u^2 + v^2 = 1$ and $\mathcal{D}\theta = \frac{u\mathcal{D}v - v\mathcal{D}u}{u^2 + v^2}$, they reformulate the gradient flow in (u, v) as

$$\begin{aligned} uu_t + vv_t &= 0, \\ \frac{uv_t - vu_t}{u^2 + v^2} &= \mathcal{D} \cdot \left(\left| \frac{u\mathcal{D}v - v\mathcal{D}u}{u^2 + v^2} \right|^{p-2} \frac{u\mathcal{D}v - v\mathcal{D}u}{u^2 + v^2} \right), \end{aligned}$$

which they then further simplify into a set of equations similar to (2.13). From this, an updating scheme similar to (2.4) is proposed. Our approach generalizes theirs for p -harmonic maps into S^2 .

In the next subsection we take a ‘‘discretize-then-optimize’’ approach. In an ‘‘optimize-then-discretize’’ approach, the updating formula (2.4), instead of the objective functional (2.12), is the starting point of the discretization. Given a specific discretization to (2.4) it is not easy to figure out the exact (discretized) objective function that the discretized version of (2.4) minimizes, so it is not clear how to specify certain algorithmic steps such as a line search and a convergence test.

Updating formula (2.4) and Theorem 2.2 provide some guidelines for designing a discretized energy function for Problem (2.12) (i.e., Problem (1.3)). If the gradient of the discretized energy function at the point \mathbf{U}^n can be represented in the form of $\mathbf{U}^n \times \mathbf{H}^n$ for some quantity $\mathbf{H}^n \in \mathbb{R}^3$, an updating scheme similar to (2.4) can be derived and the point-wise constraint $|\mathbf{U}^n| = 1$ can be maintained at all the iterations.

2.3. A Finite Difference Discretization Scheme. For simplicity, we assume that the domain Ω is a rectangle and discretize it as the grid:

$$(2.14) \quad \Omega_h \stackrel{def}{=} \{(x_i, y_j) \mid x_i = ih_x, y_j = jh_y, i = 0, 1, \dots, m; j = 0, 1, \dots, n\},$$

where h_x, h_y are the grid widths in the x and the y directions, respectively. We let $u_{i,j} = u(x_i, y_j)$, $v_{i,j} = v(x_i, y_j)$ and $w_{i,j} = w(x_i, y_j)$ at each grid point (x_i, y_j) . A key aspect of our discretization is the use of the mean operators $\zeta_x u$ and $\zeta_y u$, which are defined as

$$\zeta_x u_{i,j} = \frac{u_{i,j} + u_{i-1,j}}{2}, \quad \zeta_y u_{i,j} = \frac{u_{i,j} + u_{i,j-1}}{2},$$

with respect to the x direction and y directions, respectively. To match the mean operators $\zeta_x u$ and $\zeta_y u$, the backward finite difference operators $\delta_x u$ and $\delta_y u$, which are defined as

$$\delta_x u_{i,j} = \frac{u_{i,j} - u_{i-1,j}}{h_x}, \quad \delta_y u_{i,j} = \frac{u_{i,j} - u_{i,j-1}}{h_y},$$

are used to approximate the partial derivatives $\mathcal{D}u$. Similar mean operators and finite difference operators are defined for v and w . As shown in the following lemma, the property $(\mathcal{D}\mathbf{U})^\top \mathbf{U} = \mathbf{0}$ for $|\mathbf{U}| = 1$ holds for the \mathbf{U} and $\mathcal{D}\mathbf{U}$ given by the means and finite differences defined above, respectively. This result plays an important role in the derivation of the Euler Lagrange equations and the cross product form of the gradient.

LEMMA 2.7. *If $|\mathbf{U}_{i,j}| = |(u_{i,j}, v_{i,j}, w_{i,j})| = 1$ for all i and j , then*

$$(2.15) \quad \zeta_x u_{i,j} \delta_x u_{i,j} + \zeta_x v_{i,j} \delta_x v_{i,j} + \zeta_x w_{i,j} \delta_x w_{i,j} = 0,$$

$$(2.16) \quad \zeta_y u_{i,j} \delta_y u_{i,j} + \zeta_y v_{i,j} \delta_y v_{i,j} + \zeta_y w_{i,j} \delta_y w_{i,j} = 0.$$

The proof is trivial.

Next, we define

$$S_{i,j}^x = (\zeta_x u_{i,j})^2 + (\zeta_x v_{i,j})^2 + (\zeta_x w_{i,j})^2, \quad S_{i,j}^y = (\zeta_y u_{i,j})^2 + (\zeta_y v_{i,j})^2 + (\zeta_y w_{i,j})^2,$$

and

$$\begin{aligned} (f_1^x)_{i,j} &= \frac{\zeta_x u_{i,j} \delta_x v_{i,j} - \zeta_x v_{i,j} \delta_x u_{i,j}}{S_{i,j}^x}, & (f_1^y)_{i,j} &= \frac{\zeta_y u_{i,j} \delta_y v_{i,j} - \zeta_y v_{i,j} \delta_y u_{i,j}}{S_{i,j}^y}, \\ (f_2^x)_{i,j} &= \frac{\zeta_x u_{i,j} \delta_x w_{i,j} - \zeta_x w_{i,j} \delta_x u_{i,j}}{S_{i,j}^x}, & (f_2^y)_{i,j} &= \frac{\zeta_y u_{i,j} \delta_y w_{i,j} - \zeta_y w_{i,j} \delta_y u_{i,j}}{S_{i,j}^y}, \\ (f_3^x)_{i,j} &= \frac{\zeta_x v_{i,j} \delta_x w_{i,j} - \zeta_x w_{i,j} \delta_x v_{i,j}}{S_{i,j}^x}, & (f_3^y)_{i,j} &= \frac{\zeta_y v_{i,j} \delta_y w_{i,j} - \zeta_y w_{i,j} \delta_y v_{i,j}}{S_{i,j}^y}, \end{aligned}$$

to approximate the terms $\frac{u\mathcal{D}v-v\mathcal{D}u}{u^2+v^2+w^2}$, $\frac{u\mathcal{D}w-w\mathcal{D}u}{u^2+v^2+w^2}$ and $\frac{v\mathcal{D}w-w\mathcal{D}v}{u^2+v^2+w^2}$, in the x and y directions, respectively. The terms $(f_k^x)_{i,j}$ and $(f_k^y)_{i,j}$ for $k = 1, 2, 3$ are well defined if $S_{i,j}^x \neq 0$ or $S_{i,j}^y \neq 0$. Furthermore, we introduce

$$F_{i,j} = ((f_1^x)_{i,j}^2 + (f_1^y)_{i,j}^2 + (f_2^x)_{i,j}^2 + (f_2^y)_{i,j}^2 + (f_3^x)_{i,j}^2 + (f_3^y)_{i,j}^2 + \xi)^{p/2},$$

where ξ ($\xi = 0$ if p is even and $\xi > 0$ if p is odd) is a small perturbation to avoid non-differentiability. Using the above definitions, the objective functional in (2.12) is discretized as:

$$(2.17) \quad E_p(\mathbf{U}) = \sum_{i=1}^m \sum_{j=1}^n F_{i,j},$$

by abusing the notation and letting $\mathbf{U} = \{(u_{i,j}, v_{i,j}, w_{i,j})\}$.

We show below in Lemma 2.8, proved in Appendix B, that the partial derivatives of the discrete function $E_p(\mathbf{U})$ with respect to $u_{i,j}$, $v_{i,j}$ and $w_{i,j}$ can be represented in a cross-product form similar to their continuous counterparts (2.3).

LEMMA 2.8. *Suppose $|\mathbf{U}_{i,j}| = 1$ for all i and j . The partial derivatives of $E_p(\mathbf{U})$ with respect to variables $\mathbf{U}_{i,j} = (u_{i,j}, v_{i,j}, w_{i,j})$ for each $i = 1, \dots, m-1$ and $j = 1, \dots, n-1$ are*

$$(2.18) \quad (\nabla E_p(\mathbf{U}))_{i,j} = \mathbf{U}_{i,j} \times \mathbf{H}_{i,j},$$

where $\mathbf{H} = (\mathbf{H}_a, \mathbf{H}_b, \mathbf{H}_c)$ and

$$(2.19) \quad \begin{cases} (\mathbf{H}_c)_{i,j} = +p \left(-F_{i,j}^{\frac{p-2}{p}} \left(\frac{(f_1^x)_{i,j}}{h_x S_{i,j}^x} + \frac{(f_1^y)_{i,j}}{h_y S_{i,j}^y} \right) + F_{i+1,j}^{\frac{p-2}{p}} \frac{(f_1^x)_{i+1,j}}{h_x S_{i+1,j}^x} + F_{i,j+1}^{\frac{p-2}{p}} \frac{(f_1^y)_{i,j+1}}{h_y S_{i,j+1}^y} \right), \\ (\mathbf{H}_b)_{i,j} = -p \left(-F_{i,j}^{\frac{p-2}{p}} \left(\frac{(f_2^x)_{i,j}}{h_x S_{i,j}^x} + \frac{(f_2^y)_{i,j}}{h_y S_{i,j}^y} \right) + F_{i+1,j}^{\frac{p-2}{p}} \frac{(f_2^x)_{i+1,j}}{h_x S_{i+1,j}^x} + F_{i,j+1}^{\frac{p-2}{p}} \frac{(f_2^y)_{i,j+1}}{h_y S_{i,j+1}^y} \right), \\ (\mathbf{H}_a)_{i,j} = +p \left(-F_{i,j}^{\frac{p-2}{p}} \left(\frac{(f_3^x)_{i,j}}{h_x S_{i,j}^x} + \frac{(f_3^y)_{i,j}}{h_y S_{i,j}^y} \right) + F_{i+1,j}^{\frac{p-2}{p}} \frac{(f_3^x)_{i+1,j}}{h_x S_{i+1,j}^x} + F_{i,j+1}^{\frac{p-2}{p}} \frac{(f_3^y)_{i,j+1}}{h_y S_{i,j+1}^y} \right). \end{cases}$$

REMARK 2.9. *Forward finite differences or central finite differences can also be used, and they give similar results with appropriate mean operators.*

2.4. A Discrete Descent Method and Curvilinear Search Algorithm.

In this subsection, we describe a method for solving the unconstrained discretized problem (2.17), based upon the special structure of the gradient (2.18). Our updating scheme is analogous to (2.4). Essentially, we define a curve from the current point \mathbf{U}^n on the surface of the sphere S^2 and search along it for a new point \mathbf{U}^{n+1} .

The steepest descent method computes a new point $\mathbf{U}^n(\tau)$ by the formula

$$(2.20) \quad \mathbf{U}_{i,j}^n(\tau) = \mathbf{U}_{i,j}^n - \tau (\nabla E_p(\mathbf{U}^n))_{i,j},$$

where $\nabla E_p(\mathbf{U}^n)$ is given by (2.18). Since in general $|\mathbf{U}_{i,j}^n(\tau)| \neq 1$, we modify (2.20) in the same way as in Theorem 2.2 and obtain

$$(2.21) \quad \mathbf{U}_{i,j}^n(\tau) = \mathbf{U}_{i,j}^n - \tau \frac{\mathbf{U}_{i,j}^n(\tau) + \mathbf{U}_{i,j}^n}{2} \times \mathbf{H}_{i,j}^n,$$

which is clearly linear in $\mathbf{U}_{i,j}^n(\tau)$. Below we derive the candidate point $\mathbf{U}^n(\tau)$ and its derivative with respect to τ and show $|\mathbf{U}^n(\tau)| = |\mathbf{U}^n|$ for any τ similar to Theorem 2.2.

THEOREM 2.10. *For any τ , the solution $\mathbf{U}_{i,j}^n(\tau)$ of the system of equations (2.21) with respect to $\mathbf{U}_{i,j}^n$ is*

$$(2.22) \quad \mathbf{U}_{i,j}^n(\tau) = (\mathcal{W}_{i,j}^{n-})^{-1} \mathcal{W}_{i,j}^{n+} \mathbf{U}_{i,j}^n,$$

where $\mathcal{W}_{i,j}^{n+} = I + \frac{\tau}{2}(\mathbf{H}_{i,j}^n)^{(\times)}$, $\mathcal{W}_{i,j}^{n-} = I - \frac{\tau}{2}(\mathbf{H}_{i,j}^n)^{(\times)}$, I is the identity matrix in $\mathbb{R}^{3 \times 3}$ and $(\mathbf{H}_{i,j}^n)^{(\times)}$ is the matrix corresponding to $(\mathbf{H}_{i,j}^n)$ defined as in (2.5). Moreover $|\mathbf{U}_{i,j}^n(\tau)| = |\mathbf{U}_{i,j}^n|$ and the derivative of $\mathbf{U}_{i,j}^n(\tau)$ with respect to τ is

$$(2.23) \quad (\mathbf{U}_{i,j}^n)'(\tau) = \frac{d\mathbf{U}_{i,j}^n(\tau)}{d\tau} = \frac{1}{2} (\mathcal{W}_{i,j}^{n-})^{-1} (\mathbf{H}_{i,j}^n)^{(\times)} (\mathbf{U}_{i,j}^n + \mathbf{U}_{i,j}^n(\tau)).$$

Proof. Except for the derivation of (2.23), the proof is identical to the proof of Theorem 2.2 with \mathbf{U}^n , \mathbf{U}^{n+1} and \mathbf{H}^n replaced by $\mathbf{U}_{i,j}^n$, $\mathbf{U}_{i,j}^{n+1}(\tau)$ and $\mathbf{H}_{i,j}^n$, respectively. To obtain (2.23) we differentiate both sides of (2.22) with respect to τ , which gives

$$-\frac{1}{2}(\mathbf{H}_{i,j}^n)^{(\times)}\mathbf{U}_{i,j}^n(\tau) + \left(I - \frac{\tau}{2}(\mathbf{H}_{i,j}^n)^{(\times)}\right) \frac{d\mathbf{U}_{i,j}^n(\tau)}{d\tau} = \frac{1}{2}(\mathbf{H}_{i,j}^n)^{(\times)}\mathbf{U}_{i,j}^n.$$

□

The matrix form of the updating formula (2.22) should be expanded to avoid inverting the matrices $\mathcal{W}_{i,j}^{n-}$ at each grid point (i, j) . To simplify the notation, we temporarily drop the subscripts (i, j) and superscript n . Let $(\mathbf{H}_a)_{i,j}, (\mathbf{H}_b)_{i,j}, (\mathbf{H}_c)_{i,j}$ be denoted by a, b, c , respectively, and recall that $\mathbf{U}^n = (u, v, w)$ and $\mathbf{U}^n(\tau) = (u(\tau), v(\tau), w(\tau))$.

COROLLARY 2.11. *The solution given by (2.22) can be expressed explicitly as*

$$\begin{aligned} u(\tau) &= \frac{4u + 4\tau bw - 4\tau cv - \tau^2 b^2 u + \tau^2 a^2 u - \tau^2 c^2 u + 2\tau^2 abv + 2\tau^2 acw}{4 + \tau^2 c^2 + \tau^2 a^2 + \tau^2 b^2}, \\ v(\tau) &= \frac{4v + 4\tau cu - 4\tau aw + \tau^2 b^2 v - \tau^2 a^2 v - \tau^2 c^2 v + 2\tau^2 cbw + 2\tau^2 abu}{4 + \tau^2 c^2 + \tau^2 a^2 + \tau^2 b^2}, \\ w(\tau) &= \frac{4w + 4\tau av - 4\tau bu - \tau^2 b^2 w - \tau^2 a^2 w + \tau^2 c^2 w + 2\tau^2 bcv + 2\tau^2 acu}{4 + \tau^2 c^2 + \tau^2 a^2 + \tau^2 b^2}. \end{aligned}$$

Proof. Since a direct verification is tedious, a symbolic ‘‘MATLAB script’’ is given in Appendix A. □

REMARK 2.12. *A curve $\mu(\tau) : \mathbb{R} \rightarrow S^2$ is a geodesic on the unit sphere S^2 if and only if*

$$(2.24) \quad \ddot{\mu} = \langle \dot{\mu}, \mu \rangle \mu.$$

It can be verified that the curve $\mathbf{U}^n(\tau)$ defined by formula (2.22) is not a geodesic since it does not satisfy (2.24).

To further our understanding, we compare the curves generated by formula (2.22) with the two curves generated by the traditional steepest descent method followed by a projection on to S^2 , called the normalization method, and by the geodesic steepest descent method on S^2 . Specifically, the curve generated by the normalization method is defined as

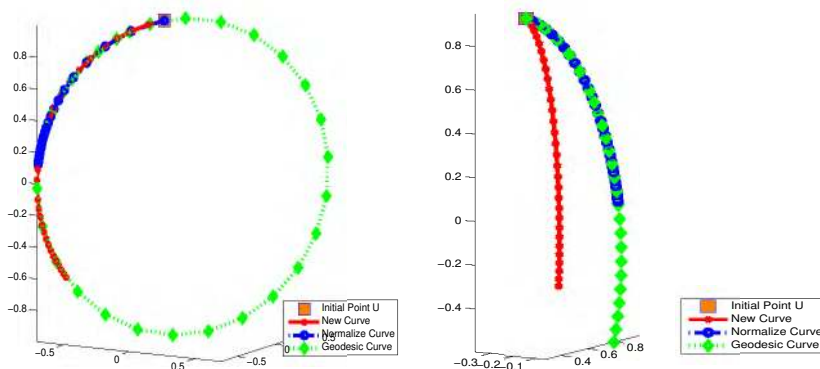
$$(2.25) \quad \widehat{\mathbf{U}}^n(\tau) = \frac{\mathbf{U}^n - \tau \nabla E_p(\mathbf{U}^n)}{|\mathbf{U}^n - \tau \nabla E_p(\mathbf{U}^n)|}.$$

The geodesic steepest descent method searches along the geodesic of S^2 in the negative gradient direction. We first make sure that the gradient $\nabla E_p(\mathbf{U}^n)$ lies on the tangent plane $T_{\mathbf{U}^n} S^2$ by computing $\mathbf{d} = \Pi_{\mathbf{U}^n}(\nabla E_p(\mathbf{U}^n))$ where $\Pi_{\mathbf{U}} \mathbf{V} = \mathbf{V} - \langle \mathbf{V}, \mathbf{U} \rangle \mathbf{U}$ is the orthogonal projection from the tangent space $T_{\mathbf{U}} \mathbb{R}^3$ onto the tangent space $T_{\mathbf{U}} S^2$. Then the curve along the geodesic of the unit sphere [Chan and Shen, 2000] is defined as

$$(2.26) \quad \widetilde{\mathbf{U}}^n(\tau) = \cos(\tau|\mathbf{d}|) \mathbf{U}^n + \sin(\tau|\mathbf{d}|) \frac{\mathbf{d}}{|\mathbf{d}|}.$$

To obtain some intuition, we study Example 4.1 with $p = 1$ from Section 4 on numerical implementation. We choose \mathbf{U}^n as the initial point \mathbf{U}^0 given by (4.1), choose a grid point (i, j) and then compute the points $\mathbf{U}^n(\tau_n)$, $\widehat{\mathbf{U}}^n(\tau_n)$ and $\widetilde{\mathbf{U}}^n(\tau_n)$ for $\tau_n = k/3$, $k = 1, \dots, 30$ on the curve $\mathbf{U}^n(\tau)$, $\widehat{\mathbf{U}}^n(\tau)$ and $\widetilde{\mathbf{U}}^n(\tau)$ defined by formulas (2.22), (2.25) and (2.26), respectively, corresponding to that grid point. The left hand side of Figure 2.1 is the result corresponding to the grid point (11, 11) and the right hand side of Figure 2.1 is the result corresponding to the grid point (11, 15). These plots show that the three curves are different.

FIG. 2.1. Comparison of Updating Formulas. The solid curve marked by * was generated by (2.22), the dash-dot curve marked by \circ was generated by the normalization method, and the dashed curve marked by \diamond was generated by the geodesic steepest descent method.



It is well known that the steepest descent method with a fixed step size may not converge. By choosing the step size wisely, we can guarantee convergence and even accelerate the speed of convergence without greatly increasing the computational cost at each iteration. One approach is to minimize the objective value $E_p(\mathbf{U})$ along the curve $\mathbf{U}^n(\tau)$ with respect to τ , i.e., to obtain an optimal τ^* by solving

$$\min_{\tau} E_p(\mathbf{U}^n(\tau)).$$

Since finding a global minimizer τ^* is computationally expensive, one is usually satisfied with an approximate minimizer such as a τ_n satisfying the Armijo-Wolfe conditions [Fletcher, 1987; Nocedal and Wright, 2006; Sun and Yuan, 2006]

$$(2.27a) \quad E_p(\mathbf{U}^n(\tau_n)) \leq E_p(\mathbf{U}^n(0)) + \rho_1 \tau_n E'_p(\mathbf{U}^n(0)),$$

$$(2.27b) \quad E'_p(\mathbf{U}^n(\tau_n)) \geq \rho_2 E'_p(\mathbf{U}^n(0)),$$

where $E'_p(\mathbf{U}^n(0))$ and $E'_p(\mathbf{U}^n(\tau_n))$ are the derivatives of $E_p(\mathbf{U}^n(\tau))$ with respect to τ at $\tau = 0$ and $\tau = \tau_n$, respectively, and $0 < \rho_1 < \rho_2 < 1$ are two parameters. To select a step size τ_n to satisfy the Armijo-Wolfe conditions (2.27a) and (2.27b), we refer to Algorithms 3.2 and 3.3 in [Nocedal and Wright, 2006], which are based on interpolation and bisection. For a more detailed description of these kind of strategies, see, for example [Moré and Thuente, 1994]. To summarize, we describe the curvilinear search approach in Algorithm 1.

Algorithm 1 A gradient descent method with curvilinear search

- STEP 0: **Initialization.** Given an initial point \mathbf{U}^0 such that $|\mathbf{U}^0| = 1$. Set $n = 0$, $\epsilon \geq 0$ and $0 < \rho_1 < \rho_2 < 1$.
- STEP 1: **Compute the Step size τ_n .** Call line search along the path $\mathbf{U}^n(\tau)$ defined by (2.22) to obtain a step size τ_n that satisfies the Armijo-Wolfe conditions (2.27a) and (2.27b).
- STEP 2: **Update.** Set the new trial point $\mathbf{U}^{n+1} = \mathbf{U}^n(\tau_n)$. If $\|\nabla E_p(\mathbf{U}^{n+1})\| \leq \epsilon$, then **STOP**. Set $n = n + 1$; goto STEP 1.
-

Using the chain rule, the derivative of $E_p(\mathbf{U}^n(\tau))$ with respect to τ is

$$(2.28) \quad E'_p(\mathbf{U}^n(\tau)) = [\nabla E_p(\mathbf{U}^n(\tau))]^\top (\mathbf{U}^n)'(\tau),$$

where the partial derivatives $(\mathbf{U}^n)'(\tau)$ are given by (2.23). Using (2.23) we have

$$(2.29) \quad (\mathbf{U}_{i,j}^n)'(0) = (\mathbf{H}_{i,j}^n) \times \mathbf{U}_{i,j}^n = -(\nabla E_p(\mathbf{U}^n))_{i,j},$$

or

$$(2.30) \quad E_p'(\mathbf{U}^n(0)) = -\|\nabla E_p(\mathbf{U}^n)\|^2 \leq 0.$$

Moreover, (2.29) shows that the negative gradient is the direction of the trajectory $\mathbf{U}^n(\tau)$ at $\tau = 0$.

Since $E_p(\mathbf{U}(\tau))$ is continuously differentiable and bounded from below, it is not difficult to prove that there exists a τ_n satisfying the Armijo-Wolfe condition (2.27a) and (2.27b). Therefore, every iteration of Algorithm 1 is well defined. Formally, we have

LEMMA 2.13 ([Nocedal and Wright, 2006]: Lemma 3.1). *If $0 < \rho_1 < \rho_2 < 1$ and $E_p'(\mathbf{U}^n(0)) < 0$, there exist intervals of step lengths satisfying the Armijo-Wolfe conditions (2.27a) and (2.27b).*

To prove the global convergence of Algorithm 1, let us define the level set

$$(2.31) \quad \mathcal{L} = \{\mathbf{U} \mid E_p(\mathbf{U}) \leq E_p(\mathbf{U}^0), |\mathbf{U}_{i,j}^0| = 1, |\mathbf{U}_{i,j}| = 1\},$$

which is a compact set. Starting from a point \mathbf{U}^0 with $|\mathbf{U}_{i,j}^0| = 1$, the sequence $\{\mathbf{U}^n\}$ generated by Algorithm 1 stays in the level set \mathcal{L} since $E_p(\mathbf{U}^n)$ is decreasing and \mathbf{U}^n always satisfies the point wise constraints $|\mathbf{U}_{i,j}^n| = 1$. We first show in Lemma 2.14 that $\lim_{n \in K} \|\mathbf{U}^n(\tau_n) - \mathbf{U}^n(0)\| = 0$ and $\lim_{n \in K} \|(\mathbf{U}^n)'(\tau_n) - (\mathbf{U}^n)'(0)\| = 0$ if $\tau_n \rightarrow 0$ for any subsequence $\{\mathbf{U}^n\}_{n \in K}$ generated by Algorithm 1.

LEMMA 2.14. *Suppose that $\{\mathbf{U}^n\}_{n \in K}$ is an infinite subsequence generated by Algorithm 1. Then the sequence $\{\mathbf{U}^n(\tau_n)\}_{n \in K}$ defined by formula (2.22) satisfies $\lim_{n \in K} \|\mathbf{U}^n(\tau_n) - \mathbf{U}^n(0)\| = 0$ and $\lim_{n \in K} \|(\mathbf{U}^n)'(\tau_n) - (\mathbf{U}^n)'(0)\| = 0$ if $\lim_{n \in K} \tau_n = 0$.*

Proof. 1. It follows from (2.22) that

$$\mathbf{U}_{i,j}^n(\tau) - \mathbf{U}_{i,j}^n(0) = \left[(\mathcal{W}_{i,j}^{n-})^{-1} \mathcal{W}_{i,j}^{n+} - I \right] \mathbf{U}_{i,j}^n = \widehat{\mathcal{W}}_{i,j}^n \mathbf{U}_{i,j}^n,$$

and hence that

$$|\mathbf{U}_{i,j}^n(\tau) - \mathbf{U}_{i,j}^n(0)| \leq \left\| \widehat{\mathcal{W}}_{i,j}^n \right\|_2 |\mathbf{U}_{i,j}^n| = \kappa_{i,j}^n,$$

where $\kappa_{i,j}^n = \frac{2\tau_n |\mathbf{H}_{i,j}^n|}{\sqrt{4 + \tau_n^2 |\mathbf{H}_{i,j}^n|^2}}$ is the 2-norm of the matrix $\widehat{\mathcal{W}}_{i,j}^n$ (See Appendix A for a procedure for determining this norm). Therefore, we obtain

$$\|\mathbf{U}^n(\tau_n) - \mathbf{U}^n(0)\|^2 = \sum_{i,j} |\mathbf{U}_{i,j}^n(\tau_n) - \mathbf{U}_{i,j}^n(0)|^2 \leq \sum_{i,j} (\kappa_{i,j}^n)^2.$$

Since $\{\mathbf{H}_{i,j}^n\}_{n \in K}$ are continuous functions with respect to the variables \mathbf{U}^n , which stays in the compact level set \mathcal{L} , it follows that the sequences $\{\mathbf{H}_{i,j}^n\}_{n \in K}$ are uniformly bounded for all i and j . This together with the fact that $\lim_{n \in K} \tau_n = 0$, gives $\lim_{n \in K} (\kappa_{i,j}^n)^2 = 0$, and hence the result $\lim_{n \in K} \|\mathbf{U}^n(\tau_n) - \mathbf{U}^n(0)\| = 0$.

2. It follows from (2.22) and (2.23) in Theorem 2.10 that

$$\begin{aligned} (\mathbf{U}_{i,j}^n)'(\tau) - (\mathbf{U}_{i,j}^n)'(0) &= \frac{1}{2} (\mathcal{W}_{i,j}^{n-})^{-1} (\mathbf{H}_{i,j}^n)^{(\times)} (\mathbf{U}_{i,j}^n + \mathbf{U}_{i,j}^n(\tau)) - (\mathbf{H}_{i,j}^n)^{(\times)} \mathbf{U}_{i,j}^n \\ &= \left[\frac{1}{2} (\mathcal{W}_{i,j}^{n-})^{-1} (\mathbf{H}_{i,j}^n)^{(\times)} \left(I + (\mathcal{W}_{i,j}^{n-})^{-1} \mathcal{W}_{i,j}^{n+} \right) - (\mathbf{H}_{i,j}^n)^{(\times)} \right] \mathbf{U}_{i,j}^n \\ &= \widehat{\mathcal{W}}_{i,j}^n \mathbf{U}_{i,j}^n, \end{aligned}$$

where the 2-norm of the matrix $\widetilde{\mathcal{W}}_{i,j}^n$ is

$$\widetilde{\kappa}_{i,j}^n = \sqrt{\frac{\tau_n^4 |\mathbf{H}_{i,j}^n|^6 + 16\tau_n^2 |\mathbf{H}_{i,j}^n|^4}{\tau_n^4 |\mathbf{H}_{i,j}^n|^4 + 8\tau_n^2 |\mathbf{H}_{i,j}^n|^2 + 16}}.$$

Therefore, a similar proof to that given in part 1 gives $\lim_{n \in K} \|(\mathbf{U}^n)'(\tau_n) - (\mathbf{U}^n)'(0)\| = 0$. \square

We now study the convergence properties of the sequence $\{\mathbf{U}^n\}$ generated by Algorithm 1. Since $\{\mathbf{U}^n\}$ stays in the compact level set \mathcal{L} , there exists at least one accumulation point. Moreover, the following result shows that the sequence of gradient $\{\nabla E_p(\mathbf{U}^n)\}$ converges to zero.

THEOREM 2.15. *Let $\{\mathbf{U}^n\}$ be the full sequence generated by Algorithm 1. Then*

$$(2.32) \quad \lim_{n \rightarrow \infty} \|\nabla E_p(\mathbf{U}^n)\| = 0.$$

Proof. For a proof by contradiction we suppose that (2.32) does not hold. Then there exists a constant $\epsilon > 0$ and a infinite index set $K \subseteq \mathbb{N}$ such that

$$(2.33) \quad \|\nabla E_p(\mathbf{U}^k)\| > \epsilon, \text{ for all } n \in K.$$

It follows from Lemma 2.13 that a step size that satisfies the Armijo-Wolfe condition (2.27a) and (2.27b) is well defined for each iteration. Summing the inequalities (2.27a) we obtain that

$$(2.34) \quad \sum_{n=0}^{\infty} \rho_1 \tau_n \|\nabla E_p(\mathbf{U}^n)\|^2 \leq E_p(\mathbf{U}^0) - \lim_{n \rightarrow \infty} E_p(\mathbf{U}^n),$$

where the limit exists because of the descent property of $E_p(\mathbf{U}^n)$ and the boundedness of \mathcal{L} . Hence, we have $\tau_n \|\nabla E_p(\mathbf{U}^n)\|^2 \rightarrow 0$, which implies that $\tau_n \rightarrow 0$ for $n \in K$ because of (2.33). Therefore, from Lemma 2.14, we have

$$(2.35) \quad \lim_{n \in K} \|\mathbf{U}^n(\tau_n) - \mathbf{U}^n(0)\| = 0 \text{ and } \lim_{n \in K} \|(\mathbf{U}^n)'(\tau_n) - (\mathbf{U}^n)'(0)\| = 0.$$

Using relation (2.30) and the curvature condition (2.27b), we obtain that, for all $n \in K$,

$$(1 - \rho_2) \|\nabla E_p(\mathbf{U}^n)\|^2 = (\rho_2 - 1) E_p'(\mathbf{U}^n(0)) \leq E_p'(\mathbf{U}^n(\tau_n)) - E_p'(\mathbf{U}^n(0)).$$

Using relation (2.28), we have

$$\begin{aligned} & E_p'(\mathbf{U}^n(\tau_n)) - E_p'(\mathbf{U}^n(0)) \\ &= \nabla E_p(\mathbf{U}^n(\tau_n))^\top (\mathbf{U}^n)'(\tau_n) - \nabla E_p(\mathbf{U}^n(0))^\top (\mathbf{U}^n)'(0) \\ &= \nabla E_p(\mathbf{U}^n(\tau_n))^\top [(\mathbf{U}^n)'(\tau_n) - (\mathbf{U}^n)'(0)] + [\nabla E_p(\mathbf{U}^n(\tau_n)) - \nabla E_p(\mathbf{U}^n(0))]^\top (\mathbf{U}^n)'(0). \end{aligned}$$

Therefore, it follows from Cauchy-Schwartz inequality that

$$(2.36) \quad (1 - \rho_2) \|\nabla E_p(\mathbf{U}^n)\|^2 \leq \|\nabla E_p(\mathbf{U}^n(\tau_n))\| \|(\mathbf{U}^n)'(\tau_n) - (\mathbf{U}^n)'(0)\| \\ + \|\nabla E_p(\mathbf{U}^n(\tau_n)) - \nabla E_p(\mathbf{U}^n(0))\| \|(\mathbf{U}^n)'(0)\|.$$

Since $\nabla E_p(\mathbf{U})$ is continuous on the compact set \mathcal{L} and recalling (2.35), we have

$$\lim_{n \in K} \|\nabla E_p(\mathbf{U}^n(\tau_n)) - \nabla E_p(\mathbf{U}^n(0))\| = 0.$$

Furthermore, $\{\|\nabla E_p(\mathbf{U}^n)\|\}_{n \in K}$ is bounded, and so is $\|(\mathbf{U}^n)'(0)\|$ since by (2.29) $\|(\mathbf{U}^n)'(0)\| = \|\nabla E_p(\mathbf{U}^n)\|$. These facts together with (2.35) imply that the right hand side in (2.36) converges to zero as $n \in K$ tends to ∞ . This, in turn, implies that $\lim_{n \in K} \|\nabla E_p(\mathbf{U}^n)\| = 0$, which contradicts (2.33). \square

REMARK 2.16. *If a backtracking line search is used in Algorithm 1, results similar to Theorem 2.15 still hold.*

3. Accelerating the Curvilinear Search Method. In this section we apply so-called Barzilai-Borwein (BB) steps [Barzilai and Borwein, 1988] for a significant acceleration of convergence. Calculating BB steps requires less computation per iteration than performing line search, and using them often significantly reduces the required number of iterations in a gradient descent method.

The BB method for solving

$$(3.1) \quad \min_{x \in \mathbb{R}^N} f(x)$$

has the form $x^{n+1} = x^n - \alpha_n g^n$, where $g^n = \nabla f(x^n)$ and α_n is determined by the information obtained at the points x^{n-1} and x^n . Let

$$s^{n-1} = x^n - x^{n-1}, \quad y^{n-1} = g^n - g^{n-1}.$$

Barzilai and Borwein choose the step size α_n so that the matrix $D^n = \alpha_n I$, which can be viewed as an approximation to the Hessian of f at x^n , has the quasi-Newton property $D^n y^{n-1} = s^{n-1}$. This yields

$$(3.2) \quad \alpha_n^1 = \frac{(s^{n-1})^\top s^{n-1}}{(s^{n-1})^\top y^{n-1}},$$

or

$$(3.3) \quad \alpha_n^2 = \frac{(s^{n-1})^\top y^{n-1}}{(y^{n-1})^\top y^{n-1}}.$$

For the discretized problem (2.17), one can apply either one of the step sizes (3.2) or (3.3), or use (3.2) and (3.3) alternatively on odd/even steps. On the very first iteration of the algorithm, s^0 and y^0 do not exist so the BB steps α_1^1 and α_1^2 are not defined and a line search must be performed. Since the steepest descent method with inexact line search usually works well in early iterations, our curvilinear search algorithm with BB steps uses this approach. The result is Algorithm 2.

Algorithm 2 A Curvilinear Search method with BB steps

- STEP 0: **Initialization.** Given an initial point \mathbf{U}^0 such that $|\mathbf{U}^0| = 1$. Run Algorithm 1 for γ steps to return another initial solution \mathbf{U}^γ , where $\gamma \geq 2$ is a prescribed integer. Set $n = \gamma + 1$, $\epsilon \geq 0$.
- STEP 1: **Compute the Step size** τ_n . If n is odd, compute τ_n by rule (3.2); otherwise, compute τ_n by rule (3.3).
- STEP 2: **Update.** Set the new trial point $\mathbf{U}^{n+1} = \mathbf{U}^n(\tau_n)$. If $\|\nabla E_p(\mathbf{U}^{n+1})\| \leq \epsilon$, then **STOP**. Set $n = n + 1$; goto STEP 1.
-

The BB method is a nonmonotone method since it does not decrease the objective value at every iteration. We note that there are many ways to improve the BB method,

incorporating it into a globalization strategy while preserving its good features. For example, a nonmonotone line search strategy that guarantees global convergence when combined with the BB method is studied in [Raydan, 1997]. An alternative step gradient method is proposed in [Dai, 2003]. For other references on the BB method, see, for example [Dai and Fletcher, 2005; Dai and Liao, 2002] and the references therein.

4. Numerical Results. In this section, we demonstrate the effectiveness of our curvilinear search algorithms on two test problems. We compared three algorithms: 1) the gradient flow method with a fixed step size, denoted by “fixed-step”, 2) Algorithm 1 with Armijo-Wolfe line search, denoted by “curve-ls”, and 3) Algorithm 2, denoted by “curve-BB”. All codes were written in MATALAB (Release 7.3.0); the curvilinear search code is based upon the code “DCSRCH” [Moré and Thunente, 1994] with an initial step size of 10^{-2} and parameters $\rho_1 = 10^{-4}$ and $\rho_2 = 0.9$ for Armijo-Wolfe conditions (2.27a) and (2.27b). In our implementation, we omitted the term h_x, h_y in the objective function. All experiments were performed on a Dell Precision 670 workstation with an Intel Xeon 3.4GHZ CPU and 6GB of RAM.

In the following examples, our tests are on mappings into S^2 .

EXAMPLE 4.1. ([Barrett et al., 2007; Bartels and Prohl, 2006c; Struwe, 1996]). Let $\Omega = (-1, 1)^2$, and the initial solution $\mathbf{U}^0 : \Omega \rightarrow S^2$ for $\mathbf{x} \in \Omega$ be defined by

$$(4.1) \quad \mathbf{U}^0(\mathbf{x}) = \left(\frac{\mathbf{x}}{|\mathbf{x}|} \sin\phi(|\mathbf{x}|), \cos\phi(|\mathbf{x}|) \right), \text{ where } \phi(r) = \begin{cases} br^2 & \text{for } r \leq 1 \\ b & \text{for } r \geq 1 \end{cases}$$

and $b = \frac{3\pi}{2}$. The Dirichlet boundary condition is taken as $\mathbf{U}(\mathbf{x}) = \left(\frac{\mathbf{x}}{|\mathbf{x}|}, 0 \right)$ on $\partial\Omega$. The grid spacing is set to $h = \sqrt{2}/2^4$. We chose three cases $p = 1, 1.5$ and 2 . We terminated all algorithms when the norm of the gradient was less than $\epsilon = 10^{-5}$ and limited the total number of iterations to 10000. For the “fixed-step” method, we set $\tau = 10^{-2}$ for $p = 1$, $\tau = 10^{-3}$ for $p = 1.5$ and $\tau = 5 * 10^{-4}$ for $p = 2$ to avoid a blow-up in the objective function. In Algorithm 2, the number of monotone curvilinear search iterations γ was set to 20.

We plot the first two components of the exact solution of the continuous problem (1.1) and the initial solution \mathbf{U}^0 in Figure 4.1. It is easy to see that the initial solution \mathbf{U}^0 is not close to the exact solution by comparing their two quiver plots at various grid points.

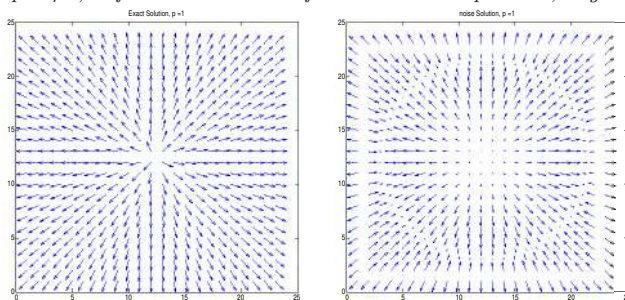
First, let us consider the $p = 1$ case (the results are in Figures 4.2 and 4.5). From Figure 4.2, all of the three methods recovered the solution quite well. The left plot in Figure 4.5 shows the energy-versus-iteration histories. All three methods reduced the objective value quickly at early iterations. Then they all slowed down, yielding “L”-shaped curves. However, it is obvious that “curve-BB” converges faster than “curve-ls”, which converges faster than “fixed-step”. The right plot in Figure 4.5 is a zoom-in of the left plot. It shows that the algorithm “curve-BB” quickly converged in no more than 200 iterations.

A summary of the computational costs for all of three methods is presented in Table 4.1, in which “CPU” denotes CPU time measured in seconds, “ITER” denotes the number of iterations, “NFE” denotes the total number of function evaluations, $E_p(\mathbf{U})$ and $\|\nabla E_p\|$ denote the objective function value and the norm of the gradient, respectively, when the algorithm terminates. From the table, the superiority of the “curve-BB” method is obvious. The “curve-BB” method took less CPU time and

TABLE 4.1
Computational Summary for Example 4.1

p	fixed-step				
	CPU (sec.)	ITER	NFE	$E_p(\mathbf{U})$	$\ \nabla E_p\ $
1	8.086515	10000	10001	7.40e+01	2.79e-05
1.5	8.101544	10000	10001	3.15e+01	4.68e-01
2	7.172569	10000	10001	1.79e+01	1.23e+00
p	curve-ls				
	CPU (sec.)	ITER	NFE	$E_p(\mathbf{U})$	$\ \nabla E_p\ $
1	5.478214	3308	3998	7.40e+01	9.90e-06
1.5	2.919280	1631	2091	3.08e+01	1.00e-05
2	1.763271	1085	1365	1.28e+01	9.92e-06
p	curve-BB				
	CPU (sec.)	ITER	NFE	$E_p(\mathbf{U})$	$\ \nabla E_p\ $
1	0.371542	331	334	7.40e+01	9.88e-06
1.5	0.283160	242	247	3.08e+01	7.90e-06
2	0.177434	162	169	1.28e+01	9.73e-06

FIG. 4.1. Example 4.1, Left: exact solution of the continuous problem, Right: initial solution



fewer function evaluations to achieve a point with a smaller gradient norm than the “curve-ls” method, which is better than the “fixed-step” method.

The recovered solution and the energy-versus-iteration histories for $p = 1.5$ are given in Figures 4.3 and 4.6 and those for $p = 2$ are given in Figures 4.4 and 4.7. These figures and Table 4.1 demonstrate that the curvilinear search methods are effective and the BB step strategy significantly accelerates convergence while the method “fixed-step” did not converge within 10000 iterations. More surprisingly, the order of the norm of the gradient achieved by the curvilinear methods is 10^{-5} while it is only of order 10^{-1} and 10^0 for the “fixed-step” method for the $p = 1.5$ and $p = 2$, respectively.

Next, we consider the application of RGB color image denoising. This involves solving Problem (1.1) with the Neumann boundary conditions $\frac{\partial \mathbf{U}}{\partial \mathbf{n}}|_{\partial \Omega} = 0$, where \mathbf{n} denotes the exterior unit normal to $\partial \Omega$. Let $\mathbf{I} = (I_R, I_G, I_B) \in \mathbb{R}^3$ be an original color image, from which we extracted the intensity or brightness $|\mathbf{I}| = \sqrt{I_R^2 + I_G^2 + I_B^2}$ and the chromaticity

$$\mathbf{f} \stackrel{\text{def}}{=} \frac{\mathbf{I}}{|\mathbf{I}|} = \left(\frac{I_R}{\sqrt{I_R^2 + I_G^2 + I_B^2}}, \frac{I_G}{\sqrt{I_R^2 + I_G^2 + I_B^2}}, \frac{I_B}{\sqrt{I_R^2 + I_G^2 + I_B^2}} \right) \in S^2.$$

Then, noise was added to the image, but only to the chromaticity \mathbf{f} so that the noisy

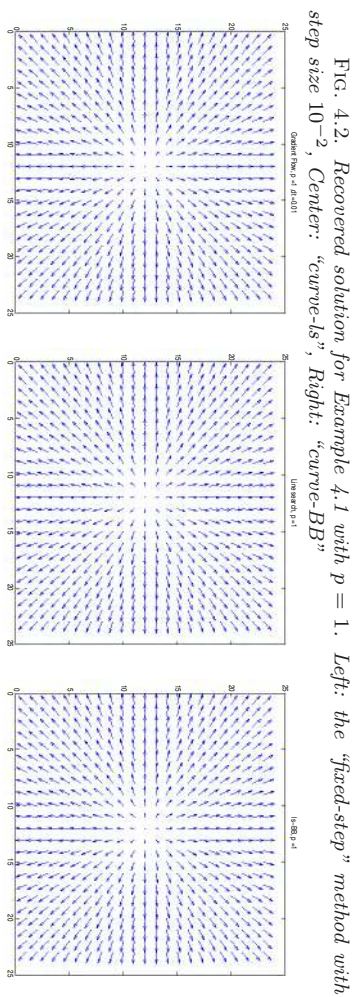


FIG. 4.2. Recovered solution for Example 4.1 with $p = 1$. Left: the “faed-step” method with step size 10^{-2} , Center: “curve- ls ”, Right: “curve- BB ”

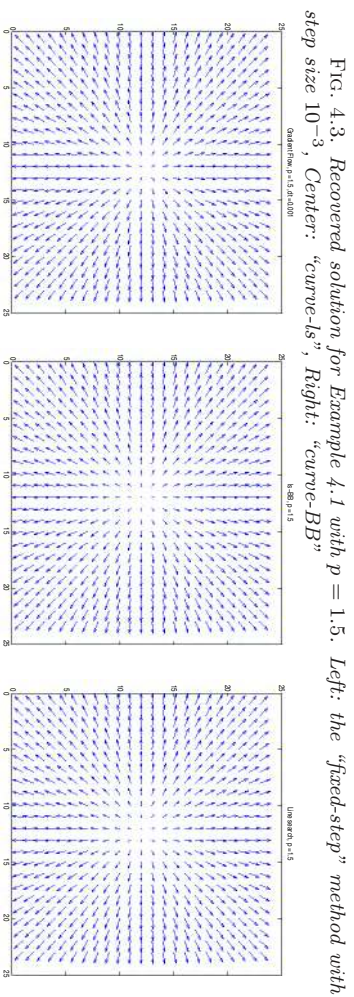


FIG. 4.3. Recovered solution for Example 4.1 with $p = 1.5$. Left: the “faed-step” method with step size 10^{-3} , Center: “curve- ls ”, Right: “curve- BB ”

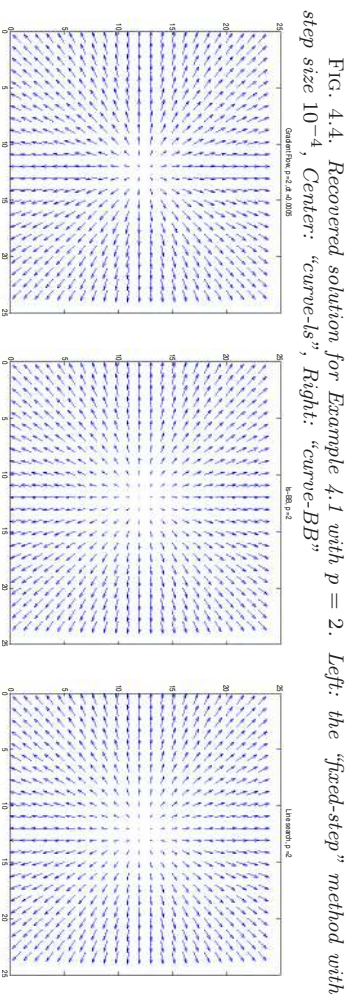


FIG. 4.4. Recovered solution for Example 4.1 with $p = 2$. Left: the “faed-step” method with step size 10^{-4} , Center: “curve- ls ”, Right: “curve- BB ”

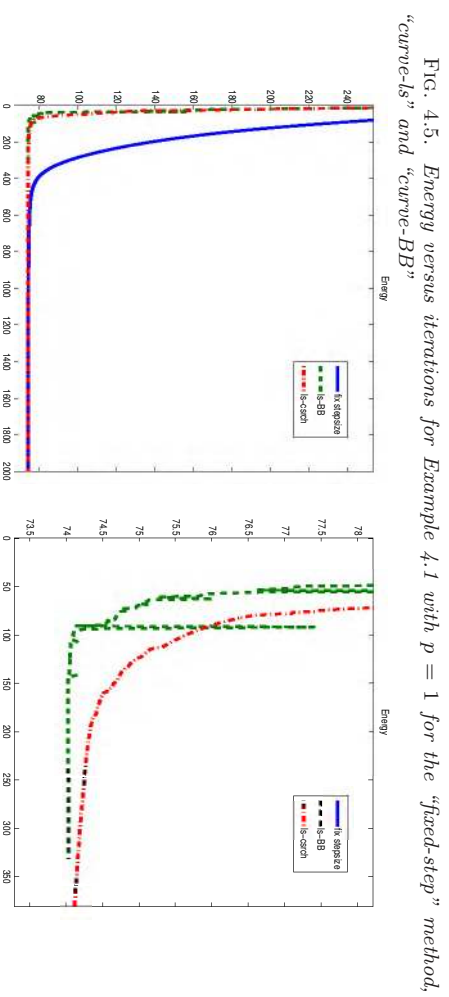


FIG. 4.5. Energy versus iterations for Example 4.1 with $p = 1$ for the “faed-step” method, “curve- ls ” and “curve- BB ”

FIG. 4.6. Energy versus iterations for Example 4.1 with $p = 1.5$ for the “fixed-step” method, ‘curve-ls’ and “curve-BB”

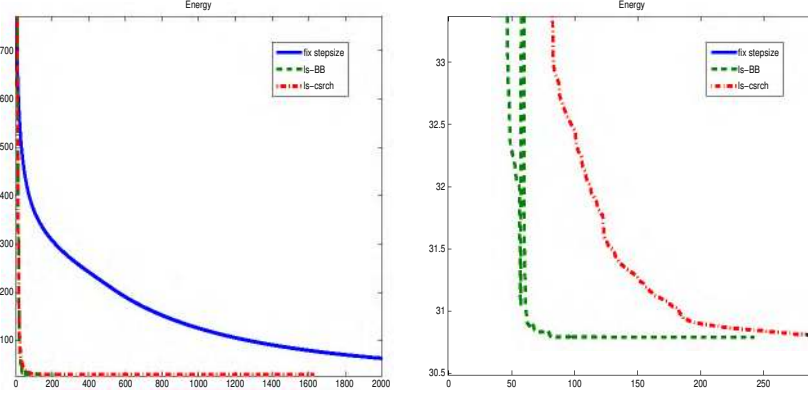
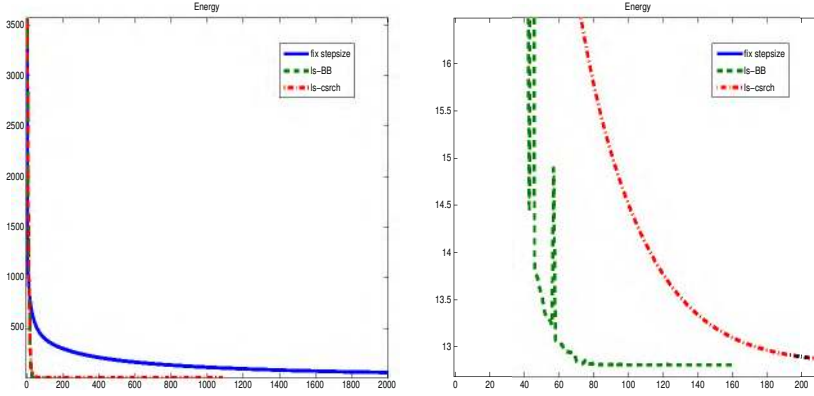


FIG. 4.7. Energy versus iterations for Example 4.1 with $p = 2$ for the “fixed-step” method, ‘curve-ls’ and “curve-BB”



chromaticity became $\mathbf{f}^{(0)} = \frac{\mathbf{f} + \beta\xi}{|\mathbf{f} + \beta\xi|}$, where $\xi \sim \text{Normal}(\mathbf{0}, \mathbf{1})$ and β was the noise level. Then applying Algorithms 1 and 2, we obtained an optimal restoration \mathbf{f}^* . Finally, using the original brightness $|\mathbf{I}|$, we assembled the new image $\mathbf{I}_{new} = \mathbf{f}^*|\mathbf{I}|$.

EXAMPLE 4.2. In Figure 4.8, we depict the original color RGB image $\mathbf{I} = (I_R, I_G, I_B) \in \mathbb{R}^3$ (left) and a noisy version (right), where the noise level β was set to 0.5. We studied two cases $p = 1$ and 2. For the fixed step size method, τ was set to 10^{-3} . We terminated the algorithm when the norm of the gradient was less than ϵ , where $\epsilon = 30$ for $p = 1$ and $\epsilon = 1$ for $p = 2$. (The solution will be over-smoothed if ϵ is too small). We also limited the total number of iterations to 500.

The denoised images are depicted in Figure 4.9 for $p = 1$ and in Figure 4.10 for $p = 2$. They show that both the “curve-BB” method and the “curve-ls” method accurately recovered the image. A summary of the computational performance for all three methods is presented in Table 4.2.

5. Conclusion. In this paper, we present new gradient descent algorithms for the p -harmonic flow problem on spheres. The algorithms are based on a simple updating formula and a specialized finite difference scheme, which preserve the point-wise constraints $|\mathbf{U}| = 1$. One of the algorithms determines a step size by an inexact curvilinear search, and is globally convergent. The other algorithm uses Barzilai-Borwein

TABLE 4.2
Computational Summary for Example 4.2

p	fixed-step				
	CPU (sec.)	ITER	NFE	$E_p(\mathbf{U})$	$\ \nabla E_p\ $
1	3.525676	500	501	4.74e+02	6.24e+01
2	2.721903	500	501	9.26e+06	1.01e+10
p	curve-ls				
	CPU (sec.)	ITER	NFE	$E_p(\mathbf{U})$	$\ \nabla E_p\ $
1	1.094670	110	117	1.78e+02	2.42e+01
2	0.487048	50	64	4.36e+00	9.31e-01
p	curve-BB				
	CPU (sec.)	ITER	NFE	$E_p(\mathbf{U})$	$\ \nabla E_p\ $
1	0.687641	66	71	2.42e+02	3.00e+01
2	0.393424	43	52	4.59e+00	9.26e-01

FIG. 4.8. Example 4.2, Left: exact solution, Right: initial solution

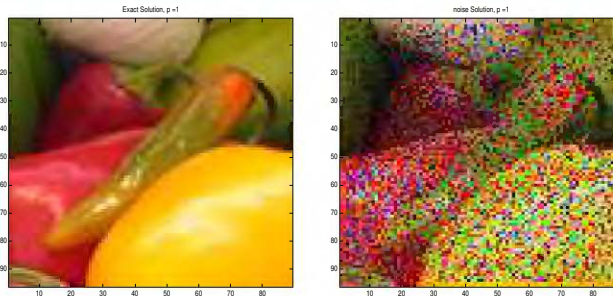


FIG. 4.9. Recovered solution for Example 4.2 with $p = 1$. Left: the “fixed-step” method with step size 10^{-3} , Center: “curve-ls”, Right: “curve-BB”

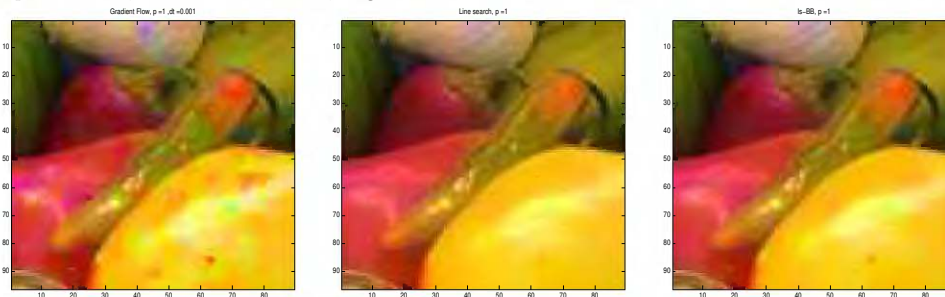
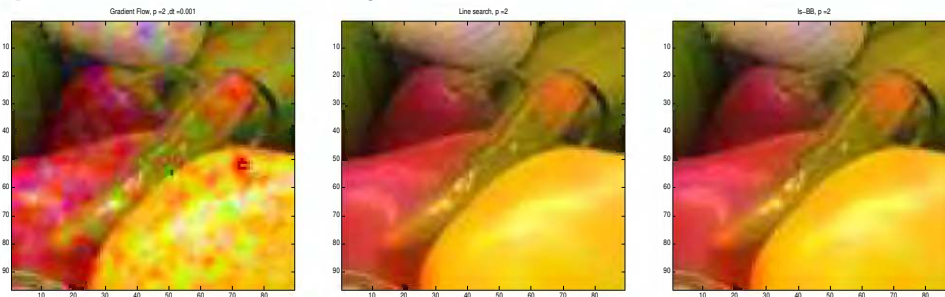


FIG. 4.10. Recovered solution for Example 4.2 with $p = 2$. Left: the “fixed-step” method with step size 10^{-3} , Center: “curve-ls”, Right: “curve-BB”



step sizes and is nonmonotonic. While not shown to converge in theory, the latter algorithm exhibits exceptional computational efficiency.

Our future work includes extending the proposed algorithmic framework to problems in higher dimensional manifolds and adapting classical optimization techniques such as Newton's method and conjugate directions within this framework.

References.

- [1] François Alouges. A new algorithm for computing liquid crystal stable configurations: the harmonic mapping case. *SIAM J. Numer. Anal.*, 34(5):1708–1726, 1997. ISSN 0036-1429.
- [2] John W. Barrett, Sören Bartels, Xiaobing Feng, and Andreas Prohl. A convergent and constraint-preserving finite element method for the p -harmonic flow into spheres. *SIAM J. Numer. Anal.*, 45(3):905–927 (electronic), 2007. ISSN 0036-1429.
- [3] Sören Bartels. Stability and convergence of finite-element approximation schemes for harmonic maps. *SIAM J. Numer. Anal.*, 43(1):220–238 (electronic), 2005. ISSN 0036-1429.
- [4] Sören Bartels and Andreas Prohl. Stable discretization of scalar and constrained vectorial perona-malik equation. Technical report, Department of Mathematics, Humboldt-Universität zu Berlin, 2006b.
- [5] Sören Bartels and Andreas Prohl. Fully practical, constraint preserving, implicit approximation of harmonic map heat flow into spheres. Technical report, Department of Mathematics, Humboldt-Universität zu Berlin, 2006a.
- [6] Sören Bartels and Andreas Prohl. Convergence of an implicit finite element method for the Landau-Lifshitz-Gilbert equation. *SIAM J. Numer. Anal.*, 44(4):1405–1419 (electronic), 2006. ISSN 0036-1429.
- [7] Sören Bartels and Andreas Prohl. Convergence of an implicit, constraint preserving finite element discretization of p -harmonic heat flow into spheres. Technical report, Department of Mathematics, Humboldt-Universität zu Berlin, 2006c.
- [8] Jonathan Barzilai and Jonathan M. Borwein. Two-point step size gradient methods. *IMA J. Numer. Anal.*, 8(1):141–148, 1988. ISSN 0272-4979.
- [9] Fabrice Bethuel, Haïm Brezis, and Frédéric Hélein. Limite singulière pour la minimisation de fonctionnelles du type Ginzburg-Landau. *C. R. Acad. Sci. Paris Sér. I Math.*, 314(12):891–895, 1992. ISSN 0764-4442.
- [10] Fabrice Bethuel, Haïm Brezis, and Frédéric Hélein. Asymptotics for the minimization of a Ginzburg-Landau functional. *Calc. Var. Partial Differential Equations*, 1(2):123–148, 1993. ISSN 0944-2669.
- [11] Thomas Cecil, Stanley Osher, and Luminita Vese. Numerical methods for minimization problems constrained to S^1 and S^2 . *J. Comput. Phys.*, 198(2):567–579, 2004. ISSN 0021-9991.
- [12] Tony Chan and Jianhong Shen. Variational restoration of nonflat image features: models and algorithms. *SIAM J. Appl. Math.*, 61(4):1338–1361 (electronic), 2000. ISSN 0036-1399.
- [13] Yunmei Chen. The weak solutions to the evolution problems of harmonic maps. *Math. Z.*, 201(1):69–74, 1989. ISSN 0025-5874.
- [14] Yunmei Chen and Fang-Hua Lin. Remarks on approximate harmonic maps. *Comment. Math. Helv.*, 70(1):161–169, 1995. ISSN 0010-2571.

- [15] Yunmei Chen and Michael Struwe. Existence and partial regularity results for the heat flow for harmonic maps. *Math. Z.*, 201(1):83–103, 1989. ISSN 0025-5874.
- [16] Robert Cohen, Robert Hardt, David Kinderlehrer, San Yih Lin, and Mitchell Luskin. Minimum energy configurations for liquid crystals: computational results. In *Theory and applications of liquid crystals (Minneapolis, Minn., 1985)*, volume 5 of *IMA Vol. Math. Appl.*, pages 99–121. Springer, New York, 1987.
- [17] Robert Cohen, San Yih Lin, and Mitchell Luskin. Relaxation and gradient methods for molecular orientation in liquid crystals. *Comput. Phys. Comm.*, 53(1-3):455–465, 1989. ISSN 0010-4655. Practical iterative methods for large scale computations (Minneapolis, MN, 1988).
- [18] Jean-Michel Coron and Robert Gulliver. Minimizing p -harmonic maps into spheres. *J. Reine Angew. Math.*, 401:82–100, 1989. ISSN 0075-4102.
- [19] Yu-Hong Dai. Alternate step gradient method. *Optimization*, 52(4-5):395–415, 2003. ISSN 0233-1934. Theory, methods and applications of optimization.
- [20] Yu-Hong Dai and Roger Fletcher. Projected Barzilai-Borwein methods for large-scale box-constrained quadratic programming. *Numer. Math.*, 100(1):21–47, 2005. ISSN 0029-599X.
- [21] Yu-Hong Dai and Li-Zhi Liao. \mathbf{R} -linear convergence of the Barzilai and Borwein gradient method. *IMA J. Numer. Anal.*, 22(1):1–10, 2002. ISSN 0272-4979.
- [22] R. Fletcher. Practical methods of optimization. pages xiv+436, 1987.
- [23] Norbert Hungerbühler. Global weak solutions of the p -harmonic flow into homogeneous spaces. *Indiana Univ. Math. J.*, 45(1):275–288, 1996. ISSN 0022-2518.
- [24] Norbert Hungerbühler. Non-uniqueness for the p -harmonic flow. *Canad. Math. Bull.*, 40(2):174–182, 1997. ISSN 0008-4395.
- [25] Norbert Hungerbühler. Heat flow into spheres for a class of energies. 59:45–65, 2004.
- [26] Martin Kružík and Andreas Prohl. Recent developments in the modeling, analysis, and numerics of ferromagnetism. *SIAM Rev.*, 48(3):439–483 (electronic), 2006. ISSN 0036-1445.
- [27] Fang-Hua Lin. A remark on the map $x/|x|$. *C. R. Acad. Sci. Paris Sér. I Math.*, 305(12):529–531, 1987. ISSN 0249-6291.
- [28] San Yih Lin and Mitchell Luskin. Relaxation methods for liquid crystal problems. *SIAM J. Numer. Anal.*, 26(6):1310–1324, 1989. ISSN 0036-1429.
- [29] Marius Lysaker, Stanley Osher, and Xue-Cheng Tai. Noise removal using smoothed normals and surface fitting. *IEEE Trans. Image Process.*, 13(10):1345–1357, 2004. ISSN 1057-7149.
- [30] Facundo Mémoli, Guillermo Sapiro, and Stanley Osher. Solving variational problems and partial differential equations mapping into general target manifolds. *J. Comput. Phys.*, 195(1):263–292, 2004. ISSN 0021-9991.
- [31] Masashi Misawa. Approximation of p -harmonic maps by the penalized equation. In *Proceedings of the Third World Congress of Nonlinear Analysts, Part 2 (Catania, 2000)*, volume 47, pages 1069–1080, 2001.
- [32] Jorge J. Moré and David J. Thuente. Line search algorithms with guaranteed sufficient decrease. *ACM Trans. Math. Software*, 20(3):286–307, 1994. ISSN 0098-3500.

- [33] Jorge Nocedal and Stephen J. Wright. *Numerical optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, second edition, 2006. ISBN 978-0387-30303-1; 0-387-30303-0.
- [34] P Perona. Orientation diffusion. *IEEE Trans. Image Process*, 79:12–49, 1998.
- [35] Marcos Raydan. The Barzilai and Borwein gradient method for the large scale unconstrained minimization problem. *SIAM J. Optim.*, 7(1):26–33, 1997. ISSN 1052-6234.
- [36] Michael Struwe. Geometric evolution problems. *Nonlinear partial differential equations in differential geometry (Park City, UT, 1992)*, 2:257–339, 1996.
- [37] Wenyu Sun and Ya-Xiang Yuan. *Optimization theory and methods*, volume 1 of *Springer Optimization and Its Applications*. Springer, New York, 2006. ISBN 978-0-387-24975-9; 0-387-24975-3. Nonlinear programming.
- [38] B Tang, G Sapiro, and V Caselles. Diffusion of general data on non-flat manifolds via harmonic maps theory: The direction diffusion case. *International Journal of Computer Vision*, 36:149–161, 2000.
- [39] B Tang, G Sapiro, and V Caselles. Color image enhancement via chromaticity diffusion. *IEEE TRANSACTIONS ON IMAGE PROCESSING*, 10(5):701–707, 2001.
- [40] Luminita A. Vese and Stanley J. Osher. Numerical methods for p -harmonic flows and applications to image processing. *SIAM J. Numer. Anal.*, 40(6):2085–2104 (electronic) (2003), 2002. ISSN 0036-1429.

Appendix A. Symbolic Computational Procedure.

Since a direct verification of Theorem 2.10 is tedious, we give a symbolic computational procedure “Listing 1” written in the language of “Matlab”. First, symbolic objects are declared to define the matrix $(\mathbf{H}_{i,j}^n)^{(\times)}$ where a, b, c denote $(\mathbf{H}_a)_{i,j}$, $(\mathbf{H}_b)_{i,j}$, $(\mathbf{H}_c)_{i,j}$, respectively. Then “Wplus” and “Wminus” denote the matrices $\mathcal{W}_{i,j}^{n+}$ and $\mathcal{W}_{i,j}^{n-}$, respectively. Finally, “Ut” is computed as the solution of (2.4), where “eye” is the function that generates an identity matrix, “inv” is the function that inverts a matrix. The function “simplify” helps us to get a simplified version of a function. We use the function “det” to check the determinant of the matrix $\mathcal{W}_{i,j}^{n-}$, so that we actually can invert it. Specifically, the command “det(Wminus)” returns $1 + 1/4 * t^2 * a^2 + 1/4 * t^2 * c^2 + 1/4 * t^2 * b^2$ which is $1 + \frac{1}{4}\tau^2 \|\mathbf{H}_{i,j}^n\|^2$ since $\|\mathbf{H}_{i,j}^n\|^2 = a^2 + b^2 + c^2$. To check the 2-norm of the matrices needed in Lemma 2.14, we declare “hatW” and “tildeW” to denote the matrices $\widehat{\mathcal{W}}_{i,j}^n$ and $\widetilde{\mathcal{W}}_{i,j}^n$, respectively. Then the function “eig” is called to compute their eigenvalues. Finally, the returned results are further simplified by noting that $\|\mathbf{H}_{i,j}^n\|^2 = a^2 + b^2 + c^2$.

LISTING 1
Verify Theorem 2.10

```

% define symbolic objects
syms a b c u v w t
% define the matrix version of the cross product
H = [0 -c b; c 0 -a; -b a 0];

% define intermediate matrix
Wminus = eye(3) - 0.5*t*H;
Wplus = eye(3) + 0.5*t*H;
```

```

% solve the system of linear equations
Ut = simplify( inv(Wminus)* Wplus * [u;v; w] )

% compute the derivative of Ut with respect t
simplify( diff(Ut, 't') )

% check the determinant of the matrix Wminus
det(Wminus)

% check the 2-norm of the matrix Lemma 2.11
hatW = inv(Wminus)* Wplus - eye(3);
simplify( eig( hatW.'*hatW) )

tiddleW = 0.5* inv(Wminus)*J*(eye(3) + inv(Wminus)* Wplus) - J;
simplify( eig( tiddleW.'*tiddleW) )

```

Appendix B. Proof of Lemma 2.8.

We now compute the gradient of the discrete energy function $E_p(\mathbf{U})$ explicitly. Since the variable $u_{i,j}$ appears only in the terms $F_{k,l}$, where (k,l) corresponds to the grid points (i,j) , $(i+1,j)$, and $(i,j+1)$, the partial derivative $\frac{\partial E_p(\mathbf{U})}{\partial u_{i,j}}$ can be written as

$$(B.1) \quad \frac{\partial E_p(\mathbf{U})}{\partial u_{i,j}} = \frac{\partial F_{i,j}}{\partial u_{i,j}} + \frac{\partial F_{i+1,j}}{\partial u_{i,j}} + \frac{\partial F_{i,j+1}}{\partial u_{i,j}}.$$

We now compute the partial derivatives of $F_{i,j}$ by using the chain rule of differentiation

$$\begin{aligned} \frac{\partial F_{i,j}}{\partial u_{k,l}} &= p F_{i,j}^{\frac{p-2}{p}} \left((f_1^x)_{i,j} \frac{\partial (f_1^x)_{i,j}}{\partial u_{k,l}} + (f_2^x)_{i,j} \frac{\partial (f_2^x)_{i,j}}{\partial u_{k,l}} + (f_3^x)_{i,j} \frac{\partial (f_3^x)_{i,j}}{\partial u_{k,l}} \right. \\ &\quad \left. + (f_1^y)_{i,j} \frac{\partial (f_1^y)_{i,j}}{\partial u_{k,l}} + (f_2^y)_{i,j} \frac{\partial (f_2^y)_{i,j}}{\partial u_{k,l}} + (f_3^y)_{i,j} \frac{\partial (f_3^y)_{i,j}}{\partial u_{k,l}} \right). \end{aligned}$$

Since $(f_1^x)_{i,j}$ is only a function of the variables $u_{i,j}$, $u_{i-1,j}$, $v_{i,j}$ and $v_{i-1,j}$ at a particular grid point (i,j) , it suffices to compute the partial derivatives of $(f_1^x)_{i,j}$ with respect to these variables; all other components are zero. We first compute $\frac{\partial (f_1^x)_{i,j}}{\partial u_{i,j}}$ explicitly as an illustration

$$\begin{aligned} \frac{\partial (f_1^x)_{i,j}}{\partial u_{i,j}} &= \frac{1}{(S_{i,j}^x)^2} \left[\left(\frac{\partial \zeta_x u_{i,j}}{\partial u_{i,j}} \delta_x v_{i,j} - \zeta_x v_{i,j} \frac{\partial \delta_x u_{i,j}}{\partial u_{i,j}} \right) S_{i,j}^x \right. \\ &\quad \left. - (\zeta_x u_{i,j} \delta_x v_{i,j} - \zeta_x v_{i,j} \delta_x u_{i,j}) \left(2 \zeta_x u_{i,j} \frac{\partial \zeta_x u_{i,j}}{\partial u_{i,j}} \right) \right]. \end{aligned}$$

Using (2.15), we obtain

$$\begin{aligned} [\zeta_x u_{i,j} \delta_x v_{i,j} - \zeta_x v_{i,j} \delta_x u_{i,j}] \zeta_x u_{i,j} &= (\zeta_x u_{i,j})^2 \delta_x v_{i,j} - \zeta_x v_{i,j} (-\zeta_x v_{i,j} \delta_x v_{i,j} - \zeta_x w_{i,j} \delta_x w_{i,j}) \\ &= (S_{i,j}^x) \delta_x v_{i,j} + (\zeta_x v_{i,j} \delta_x w_{i,j} - \zeta_x w_{i,j} \delta_x v_{i,j}) \zeta_x w_{i,j}. \end{aligned}$$

Together with $\frac{\partial \zeta_x u_{i,j}}{\partial u_{i,j}} = \frac{1}{2}$, we have

$$\begin{aligned} \frac{\partial (f_1^x)_{i,j}}{\partial u_{i,j}} &= \frac{1}{(S_{i,j}^x)^2} \left[- \left(\frac{\partial \zeta_x u_{i,j}}{\partial u_{i,j}} \delta_x v_{i,j} + \zeta_x v_{i,j} \frac{\partial \delta_x u_{i,j}}{\partial u_{i,j}} \right) S_{i,j}^x \right. \\ &\quad \left. - (\zeta_x v_{i,j} \delta_x w_{i,j} - \zeta_x w_{i,j} \delta_x v_{i,j}) \zeta_x w_{i,j} \right]. \end{aligned}$$

It follows from the fact $\frac{\partial \zeta_x u_{i,j}}{\partial u_{i,j}} \delta_x v_{i,j} + \zeta_x v_{i,j} \frac{\partial \delta_x u_{i,j}}{\partial u_{i,j}} = \frac{1}{h_x} v_{i,j}$ that

$$(B.2) \quad \frac{\partial (f_1^x)_{i,j}}{\partial u_{i,j}} = -\frac{1}{h_x S_{i,j}^x} v_{i,j} - \frac{1}{S_{i,j}^x} (f_3^x)_{i,j} \zeta_x w_{i,j}.$$

Similarly, we obtain

$$(B.3) \quad \frac{\partial (f_2^x)_{i,j}}{\partial u_{i,j}} = -\frac{1}{h_x S_{i,j}^x} w_{i,j} + \frac{1}{S_{i,j}^x} (f_3^x)_{i,j} \zeta_x v_{i,j}.$$

Since only the term $S_{i,j}^x$ in $(f_3^x)_{i,j}$ is related to $u_{i,j}$, a direct calculation gives

$$(B.4) \quad \frac{\partial (f_3^x)_{i,j}}{\partial u_{i,j}} = -\frac{1}{S_{i,j}^x} (f_3^x)_{i,j} \zeta_x u_{i,j}.$$

Noting that $-(f_1^x)_{i,j} (f_3^x)_{i,j} \zeta_x w_{i,j} + (f_2^x)_{i,j} (f_3^x)_{i,j} \zeta_x v_{i,j} = (f_3^x)_{i,j}^2 \zeta_x u_{i,j}$ and combining (B.2), (B.3) and (B.4) together, we obtain

$$(B.5) \quad (f_1^x)_{i,j} \frac{\partial (f_1^x)_{i,j}}{\partial u_{k,l}} + (f_2^x)_{i,j} \frac{\partial (f_2^x)_{i,j}}{\partial u_{k,l}} + (f_3^x)_{i,j} \frac{\partial (f_3^x)_{i,j}}{\partial u_{k,l}} = -\frac{1}{h_x S_{i,j}^x} ((f_1^x)_{i,j} v_{i,j} + (f_2^x)_{i,j} w_{i,j}).$$

Similar to (B.5), we obtain for the y direction that

$$(B.6) \quad (f_1^y)_{i,j} \frac{\partial (f_1^y)_{i,j}}{\partial u_{k,l}} + (f_2^y)_{i,j} \frac{\partial (f_2^y)_{i,j}}{\partial u_{k,l}} + (f_3^y)_{i,j} \frac{\partial (f_3^y)_{i,j}}{\partial u_{k,l}} = -\frac{1}{h_y S_{i,j}^y} ((f_1^y)_{i,j} v_{i,j} + (f_2^y)_{i,j} w_{i,j}).$$

Finally, by symmetry of the variables and noting the relationship (B.1), we can write out the gradient explicitly as

$$(B.7) \quad \begin{cases} \frac{\partial E_p(\mathbf{U})}{\partial u_{i,j}} = +v_{i,j}(\mathbf{H}_c)_{i,j} - w_{i,j}(\mathbf{H}_b)_{i,j}, \\ \frac{\partial E_p(\mathbf{U})}{\partial v_{i,j}} = -u_{i,j}(\mathbf{H}_c)_{i,j} + w_{i,j}(\mathbf{H}_a)_{i,j}, \\ \frac{\partial E_p(\mathbf{U})}{\partial w_{i,j}} = +u_{i,j}(\mathbf{H}_b)_{i,j} - v_{i,j}(\mathbf{H}_a)_{i,j}, \end{cases}$$

which leads to the cross product expression in Lemma 2.8.